IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.201X.DOI

A Hybrid Framework for Sentiment Analysis using Genetic Algorithm based Feature Reduction

FARKHUND IQBAL[1], JAHANZEB MAQBOOL[2], BENJAMIN C. M. FUNG[3], RABIA BATOOL[1], ASAD MASOOD KHATTAK[1], SAIQA ALEEM[1], AND PATRICK C. K. HUNG[4]

[1]College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates
[2]School of Electrical Engineering and Computer Science, NUST, Pakistan
[3]School of Information Studies, McGill University, Canada
[4]University of Ontario Institute of Technology, Canada

Corresponding author: Farkhund Iqbal (e-mail: Farkhund.Iqbal@zu.ac.ae).

ABSTRACT Due to the rapid development of Internet technologies and social media, sentiment analysis has become an important opinion mining technique. Recent research work has described the effectiveness of different sentiment classification techniques ranging from simple rule-based and lexicon-based approaches to more complex machine learning algorithms. While lexicon-based approaches have suffered from the lack of dictionaries and labeled data, machine learning approaches have fallen short in terms of accuracy. This paper proposes an integrated framework which bridges the gap between lexicon-based and machine learning approaches to achieve better accuracy and scalability. To solve the scalability issue that arises as the feature-set grows, a novel Genetic Algorithm (GA) based feature reduction technique is proposed. By using this hybrid approach, we are able to reduce the feature-set size by up to 42% without compromising the accuracy. The comparison of our feature reduction technique with more widely used Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA) based feature reduction techniques have shown up to 15.4% increased accuracy over PCA and up to 40.2% increased accuracy over LSA. Furthermore, we also evaluate our sentiment analysis framework on other metrics including precision, recall, F-measure, and feature size. In order to demonstrate the efficacy of GA based designs, we also propose a novel cross-disciplinary area of geopolitics as a case study application for our sentiment analysis framework. The experiment results have shown to accurately measure public sentiments and views regarding various topics such as terrorism, global conflicts, social issues etc. We envisage the applicability of our proposed work in various areas including security and surveillance, law-and-order, and public administration.

INDEX TERMS Classifier, Feature optimization, Genetic Algorithm, Machine Learning, Sentiment Analysis

I. INTRODUCTION

THE Internet and associated web technologies have dramatically changed the way our society works [1]. Social networks such as Facebook and Twitter have become commonplace for exchanging ideas, sharing information, promoting business and trade, running political and ideological campaigns, and promoting products and services [2]. Social media is generally studied from different perspectives i.e., collecting business intelligence for products and services promotion, monitoring malicious activities for detecting and mitigating cyber-threats, and sentiment analysis for analyzing people's feedback and reviews. Sentiment analysis, often referred as opinion mining, is the extraction, identification, or characterization of the sentiment from text using Natural Language Processing (NLP), statistics, or machine learning (ML) methods [3]. The field of sentiment analysis has been widely studied by researchers during the last few years [4] [5]. In this context, several approaches have been proposed, developed, and tested [3]. The most common approach is ML which needs a significant dataset for training and learning the association between different aspects and sentiments. Furthermore, ML-based models usually target a simple global classification, rather than individual aspects of the reviewed product. There are three major techniques being used for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

sentiment analysis; ML, lexicon-based, and rule-based approach [6]. ML methods use different learning algorithms and labeled dataset to train the classifier and to determine the sentiment [7]. The lexicon-based approach involves calculating the sentiment polarity of text using the semantic orientation of words or sentences [8]. The semantic orientation is a measure of subjectivity and opinion in text. The rule-based approach looks for opinion words in the text and then classifies it based on the number of positive and negative words [9]. It considers different rules for classification such as dictionary polarity, booster words, negation words, idioms etc.

Sentiment analysis is mostly discussed in the context of product reviews like; Is this product review positive or negative? Are customers satisfied or dissatisfied? Furthermore, it also helps to answer the Business Intelligence related questions like; Why aren't consumers buying our product? However, cross-domain insights and applications of sentiment analysis are scarce [10]. The examples of such applications include analysis of user opinion on the politics, sociology, and the psychology of society.

The existing research on sentiment analysis focuses on three different approaches individually. Thus, it is evident that there is a wide gap in terms of integrated tools and techniques for sentiment analysis which allow users to plug, play, and test different algorithms and optimizations based on customized preferences and parameters. In the light of this discussion, we clearly see a growing need for an integrated sentiment analysis tool which should fill the gap presented in the previous research.

This paper proposes a hybrid approach to sentiment analysis which employs state-of-the-art ML algorithms and lexical databases to automatically analyze archives of online documents (e.g., reviews, chats, and social media data). We propose a novel Genetic Algorithm (GA) based solution to feature reduction problem by developing a customized fitness function. The fitness function utilizes SentiWordNet [11] lexicon to calculate the polarity difference between a class label and feasible feature vector (potential solution). To the best of our knowledge, we are the first to employ such a hybrid approach with GA based optimized feature selection. This evolutionary approach for optimal feature selection results in increased accuracy and better scalability. The customized fitness function shows up to 42% reduced feature-set without any compromise on overall accuracy. Furthermore, in order to demonstrate the feasibility of the proposed feature reduction algorithm, we also perform a detailed comparison with other feature reduction algorithms including PCA [12] and LSA [13] which results in our system having up to 15.4% increased accuracy over PCA and up to 40.2% increased accuracy over LSA. PCA is a dimensionality reduction procedure that simplifies the complexity in high-dimensional data by reducing a large set of variables to a small set that still retains information and trends present in data. It projects a set of points onto a smaller dimensional affine subspace of "best fit". LSA is a method used in NLP that discovers a data representation which has a lower dimension than the original semantic space by analyzing relationships between documents and its terms. It decreases the dimension using a mathematical technique called singular value decomposition (SVD).

The second contribution of this work lies in the novelty of the proposed application area in the geopolitical context. There is a lack of modern sentiment analysis tools which provide insights into the cross-disciplinary domain of geopolitics. Hence available insights about people's opinions on social media, magnified in a political context, and their impact on several uprisings in parts of the world are scarce. The notable examples of such uprisings include London Riots, Occupy Wall Street, the Egyptian revolution, and Arab Spring. We aim to cater this problem by discussing our proposed framework in the context of user opinion in association with geopolitical uprisings or conflicts. The proposed framework classifies user's opinions based on their political affiliations. In addition to that, the extracted sentiments can be used for cyber-intelligence [14]. This could also be helpful in rooting out any foreign element involved in assisting the local uprisings, hence making it beneficial for the security agencies. An interesting implication of sentiment analysis and opinion mining would help governments to keep a watch on the growing trends of any political uprising. It can also be helpful in the forensic investigation of criminals, identity tracing, and criminal networks mining.

The major contributions of our work as follows;

- We design, develop, and evaluate a hybrid sentiment analysis framework by combining ML and lexicon-based approaches in order to solve the limitations of each method.
- We propose a novel feature reduction algorithm by employing a GA based approach with a customized fitness function. The fitness function utilizes SentiWordNet to evaluate feasible solutions which result in improved system scalability.
- We analyze our proposed method which shows improved accuracy as compared to the state-of-the-art feature reduction algorithms.
- We propose a novel application area of cross-disciplinary geopolitical analysis as a case study application to our framework to measure public sentiments and views regarding various topics such as terrorism, global conflicts, social issues etc.

To show the results of the proposed approach, a series of experiments is performed using three different types of dataset. One is UCI ML Repository's Sentiment Analysis dataset [15] which consists of reviews data from IMDB, Amazon, and Yelp. Second data is Twitter dataset from [16] while the third dataset is a geopolitical dataset related to 2016 United States Presidential Election [17]. The evaluation is based on several parameters including precision, recall, F-measure, scalability, and accuracy. Furthermore, we have also provided a run-time analysis of our GA based feature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

reduction algorithm.

The remainder of this paper is structured as follow. Section II presents the related work in the area of sentiment analysis, text mining, and forensic investigation. Section III consists of the proposed methodology and framework design. Experimental design and discussion are provided in Section IV. Section V presents the conclusion and possible future work.

II. RELATED WORK

In this section we discuss the prominent related research being carried out in the area of sentiment analysis and text mining. Our comparison criteria is based on the two factors we discussed before; integration of sentiment analysis approaches in a unified way and a cross-disciplinary application area. We are interested to see how user's opinion and his/her social behavior can be helpful in analyzing the current geopolitical situation and uprising.

Medhat et al. [18] presented a comprehensive overview of the recently proposed algorithms, enhancements, and applications in the area of sentiment analysis. They also discussed the related fields to sentiment analysis e.g., transfer learning, emotion detection, and building resources. They tried to give a full image of the sentiment analysis techniques and related fields with brief details. Khan et al. [19] proposed a rule-based domain-independent method which classifies subjective and objective sentences from reviews and blog comments. SentiWordNet is used to calculate the score and to determine the polarity. They showed that their proposed method is effective and it outperforms ML-based methods with an accuracy of 76.8% at the feedback level and 86.6% at the sentence level. Our proposed approach is aligned with these studies as we are also focusing on ML and lexicon-based methods. However, we are employing GA based optimized feature selection for training ML algorithms.

Agarwal et al. [20] examined sentiment analysis on Twitter data. They introduced POS-specific prior polarity features and explored the use of a tree kernel to obviate the need for tedious feature engineering. Their new features and the tree kernel performed almost at the same level and both outperformed the state-of-the-art baseline techniques. Kouloumpis et al. [21] investigated the utility of linguistic features for detecting the sentiment of Twitter messages. They evaluated the usefulness of the existing lexical resources as well as the creative language used in microblogging. Devies et al. [4] presented a language-independent model for sentiment analysis for short text forms e.g., social networks statuses. They used Twitter datasets to model happy and sad sentiments and showed that their system performed 10% better than Naive Bayes (NB) model. These three papers are employing sentiment analysis on short-text data i.e., SMS, tweets etc.

Similarly, Pontiki et al. [22] described the aspect based sentiment analysis. They identified the aspects of given target entities and the sentiment expressed for each aspect. They used manually annotated reviews of restaurants and laptops as a dataset. Njolstad et al. [23] proposed, defined, and evaluated four different feature categories composed of 26 article features for sentiment analysis. They used five different ML methods to train sentiment classifier of Norweign financial internet news articles. They achieved classification precision up to 71%. When comparing ML classifiers, they found that J48 yielded the highest performance closely followed by Random Forest (RF). We have also presented a similar comparison in which we compared different classifiers and their accuracy on our system. However, we extended our evaluation by including GA optimized features in comparison.

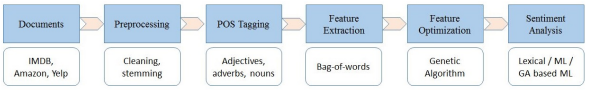

FIGURE 1. Sentiment Analysis Framework Pipeline

Govindarajan et al. [24] proposed a hybrid classification method based on integration classification methods using arcing classifier. They analyzed the performance in terms of accuracy. They designed classifier ensemble using NB and GA. They evaluated the effectiveness of ensemble technique for sentiment analysis. Finally, they evaluated the performance under different performance metrics using movie reviews datasets. However, they do not compare the performance of different classifiers and do not provide any optimization for feature size reduction.

As we observe that most of the related work employed independent techniques for sentiment analysis while using few evaluation metrics. Furthermore, they do not provide the user with the freedom to choose different algorithms, classifiers, and optimizations according to customized needs. In contrast, our proposed framework bridges the gap between sentiment analysis and geopolitical intelligence by providing 1) a unified framework having the facility to plug different algorithms, cross-validation, and optimized feature selection 2) a two-dimensional analysis on public opinions in association with political uprisings by combining security and opinion mining.

III. PROPOSED FRAMEWORK

In this work, a unified framework has been developed which includes all the components required in sentiment analysis. This modular method provides different approaches to sentiment analysis with a focus on optimizations.

The proposed framework consists of different modules which govern the internal working of the system. In order to automate the entire framework, we employ a pipeline based approach in which different modules ranging from data cleaning, preprocessing, GA, feature generation and selection, and sentiment analysis are performed in pipeline fashion. Figure 1 explains the sentiment analysis pipeline of the whole framework. There are mainly three stages of the framework; data cleaning, data pre-processing, and analysis engine. The algorithms and the internal working of each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

module are explained in the following section.

Definition 1: Polarity Score: The sentiment score of a given word as determined by SentiWordNet ontology. The score is from 0 to 1.0 ranging from extremely negative to extremely positive sentiment. ■

A. DATA CLEANING

Data cleaning is the first module in the processing pipeline of this framework. In this phase, extracted data is streamed from the files and saved in the memory for cleaning purpose. This stage consists of three sub-stages.

1) Garbage removal

In this step, unwanted characters (non-ASCII characters) including URLs, web addresses, and online links are removed from the text using customized regular expressions.

```
/* Removes slang from a given text
 */
Input: T: text from file
output: τ: updated text
T ← T.toLowerCase();
/* simple string tokenizer          */
String[] L ← T.split(" ")
/* get slangs from dictionary       */
Set < String > slangKey ← slangs.keySet()
foreach t_i ∈ L do
    if slangKey contains t_i then
        /* update the token in text  */
        t_i ← slangs.get(t_i)
    end
end
/* update list                      */
foreach t_i ∈ L do
    τ = t_i + " "
end
return τ
```

Algorithm 1: SLANG_REMOVAL

2) Slang correction

This step involves correcting any slang and abbreviated word that is used in online conversations. We use predefined dictionaries and maps to translate slangs or abbreviation to their original and abbreviated form. e.g."ttyl" to "talk to you later" and "afk" to "away from keyboard". This is helpful for later stages because, during sentiment analysis, the abbreviated words make no sense for analysis engine. The working of this module is explained in Algorithm 1.

3) Stopword removal

Stopword removal removes very common words of a language e.g., "an", "about", "above" etc. These words usually have no impact on NLP. We use CMU's Rainbow stopword list [25] for finding any stopword in the data.

B. PREPROCESSING

This module includes different NLP tasks i.e., tokenization, word stemming, and part-of-speech tagging.

1) Tokenization

Tokenization is the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens. In order to tokenize the text, LingPipeTokenizer from Apache Lucene package [26] is used which preserves punctuations. Initially, we used StringTokenizer but due to the inherent limitations of this tokenizer, we opted for much better LingPipeTokenizer. An important point to mention is that custom data structures are designed to hold tokens (Keyword) and sentences (list of Keywords) of each document.

2) Stemming

Stemming is the process of reducing inflected word to its base or root word. The framework use porter-2 algorithm [27] to convert each token to its stem form and store in the Keyword object alongside the original token.

3) POS-Tagging

POS tagging is the process of tagging a word in a text as corresponding to a particular part of speech, based on both, its definition and its context. In order to get part-of-speech tags of the words, we use Maxent Tagger from Stanford CoreNLP [28]. Each Keyword object contains an original token, its stem form, and a pos tag associated with this token. Once the data is preprocessed, it is sent to the next module in the pipeline.

C. ANALYSIS ENGINE

This is the most vital module of the framework. It includes all the natural language based techniques for sentiment analysis. Each sentence (list of Keywords) is fed to the analysis engine and it produces the aggregated sentiment polarity score of the sentence based on different sentiment analysis techniques including lexicon-based, ML using bag-of-words as features, and hybrid approach with feature reduction using GA. A complete architecture of our system is shown in Figure 2. We will explain these approaches in detail in the following subsections.

1) Lexicon-based sentiment analysis

In this approach, after preprocessing the data, the polarity score of each token in the document is calculated. In order to calculate the polarity score, the framework uses SentiWordnet lexical database. Furthermore, the score of all the tagged keywords is aggregated on a document level to find the global score and a sentiment value of either positive "P" or negative "N" is assigned. The algorithm for sentiment scoring using SentiWordnet is described in Algorithm 2. The lexicon-based approaches which have proved to have higher accuracy are, however, limited in terms of the size of lexical databases i.e., WordNet and SentiWordNet. This

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

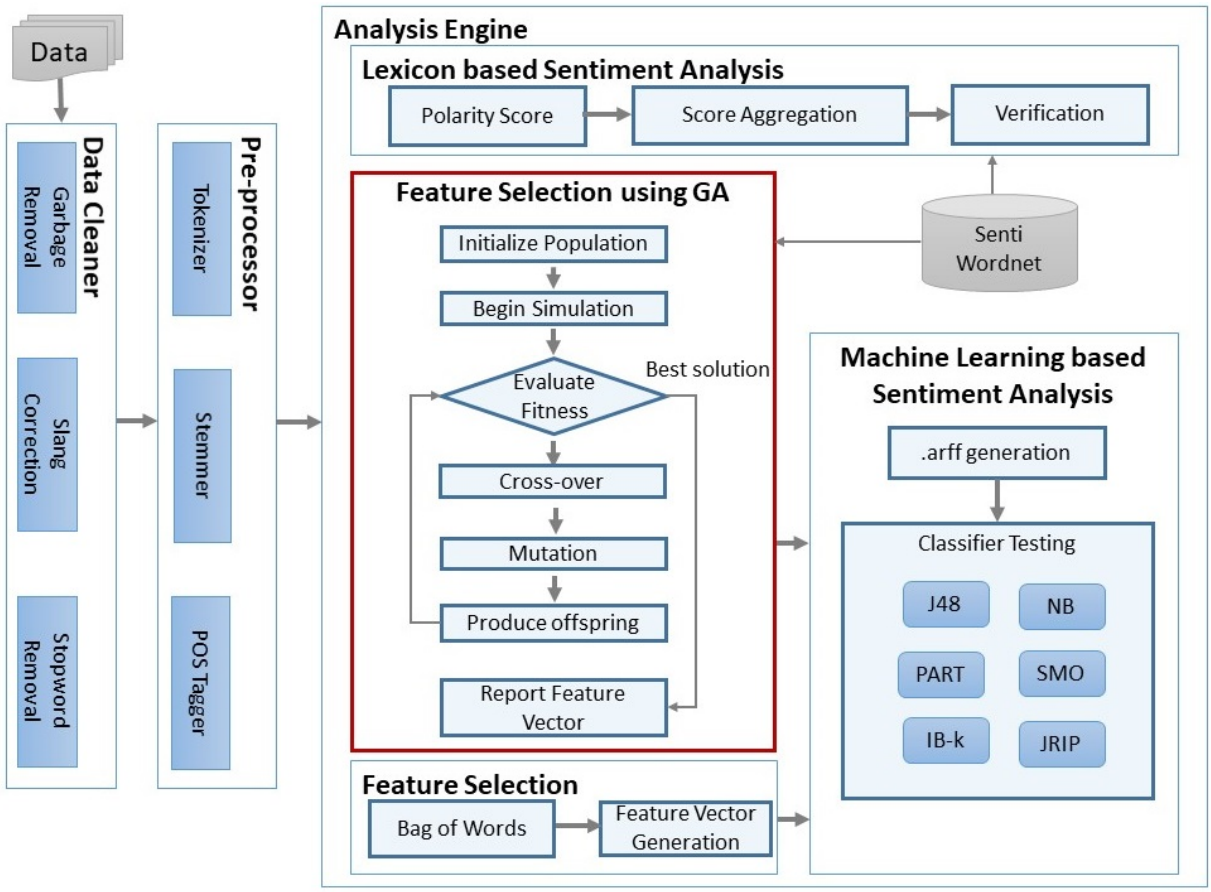

FIGURE 2. Proposed Sentiment Analysis Framework Architecture

```
/* calculates aggregated polarity
 score of a sentence                    */
```
Input: S: Sentence (a list of keywords)
output: P: Aggregated polarity score
$sum \leftarrow 0$
foreach $T_i \in S$ **do**
  $TT_i \leftarrow getPosTag(T_i)$
  $score \leftarrow getSentiWordnetScore(T_i, TT_i)$
  $sum+ = score$
end
return sum

Algorithm 2: POLARITY_SCORING_SWN

is a potential drawback which we have aimed to cater by employing a hybrid approach of lexicon-based and ML to offset the limitations.

2) ML using bag-of-words as features

In this approach, we mainly use different ML algorithms to classify sentiment values of given data. For this purpose, Weka toolkit is used because it contains several classifiers algorithms and its richness in terms of the analysis. We start by modeling our preprocessed data for Weka classifiers. In order to model the preprocessed data (list of tokens) and generated feature vector, we employ a bag-of-words approach. This is a basic approach in which we include all the potential keywords in the feature vector. We start by reading each document and add its keywords in a feature-set. Then, we append sentiment value associated with that document as a class label and generate an `ARFF` file. Finally, we process this `ARFF` file in Weka toolkit and run prominent classifier algorithms including J48, NB, PART, Sequential Minimal Optimization (SMO), Instance-Based with k-nearest neighbors (IB-k), and JRip. Here is a description of these classifiers.

- J48: J48 is a decision tree classifier in which an attribute is selected based on information gain from the training data to build each node of the tree. The selected attributes effectively split a set of training data into subsets enriched in one class or the other. It is mostly used because of its simplicity in explanation and interpretation.
- NB: It is a classification technique based on Bayes Theorem. It works with the assumption that all the attributes on the training samples are independent. It is fast and can be used with the small amount of training data. Although it is very simple, it has outperformed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

many sophisticated classification methods.

- PART: PART is a rule-based classification algorithm which generates a set of rules according to the divide-and-conquer strategy, removes all instances from the training collection that are covered by this rule and proceeds recursively until no instance remains.
- SMO: SMO is used to solve the quadratic programming problem arise in SVM training by breaking the problem into a series of smallest possible problems. Many optimizations are designed to achieve the speed up and algorithm convergence.
- IBk: IBk is among the simplest of all ML algorithms used for classification and regression predictive problems. It is a great choice for classification problems when there is little or no prior knowledge about the distribution data.
- JRip: JRip (RIPPER) is one of the basic and most popular algorithms. It implements a propositional rule learner and reduces the error using the repeated incremental pruning.

The detailed analysis is discussed in the results section.

3) Hybrid method with optimal feature selection

In this approach, we use ML algorithms to classify sentiment values of the given data. However, the problem with the previous bag-of-words approach is that it does not scale well since almost 80% of the input data gets included in the feature-set. This problem worsens as the size of the dataset grows bigger. In order to solve this scalability problem, we have devised an efficient technique to reduce the feature-set size.

We propose an evolutionary Genetic Algorithm based approach to evaluate each document and instead of choosing all the keywords, choose a subset of keywords such that the discarded keywords do not impact the overall sentiment score of the document. In other words, we aim to reduce the feature-set size by extracting those keywords that contribute towards the sentiment score of the entire document while excluded keywords make no effect. Once the feature selection is optimized, we use this feature-set to generate `ARFF` file and consequently perform the analysis using ML classifiers.

Definition 2: Chromosome (genotype): A set of parameters which define a proposed solution to the problem that the GA is trying to solve. A chromosome represents a candidate solution.■

Definition 3: Population: A set of chromosomes (candidate solutions) that evolves towards a better solution over the certain generations in order to solve the problem. Different genetic operators e.g., mutation, crossover are applied to a population.■

Definition 4: Fitness Function: The core of an evolutionary algorithm. It is a particular type of objective function which is responsible for performing the evaluation and returning a "fitness value" that reflects how optimal the solution is. The fitness value is used to determine which candidate solution (chromosome) will be surviving in the next generation.■

D. FEATURE OPTIMIZATION

Extracting features using bag-of-words data structures results in significantly large feature vector size because all the keywords which have any associated sentiment value are included in the feature vector. This technique, however, poses significant scalability problem when using a larger dataset. To solve this problem, we need to optimize the feature vector by reducing its size while maintaining accuracy. In this section, we formulated this problem and proposed its solution by using evolutionary Genetic Algorithmic approach.

1) Problem Formulation

Let `W` be a set of all the possible keywords of a document. We are interested to find a subset `S`, to be added in feature-set, of `W` which should give us a polarity value equal (or closest) to the labeled sentiment value without affecting the accuracy. This is important for the scalability of the program because if we include all the possible features then the feature vector of larger documents will grow big enough that it would not fit in the memory. Hence, in order to solve this problem, we need to optimize the feature selection technique.

The selection of a set of minimum number of features from the larger feature-set is an optimization problem with local minima. As we have discussed before, GA, due to its evolutionary nature is a well-suited technique for such non-polynomial time problems.

2) Mathematical Model

We first start with modeling our problem on an evolutionary model of GA. Let `W` be the set of all the tokens in a document after preprocessing and τ be the labeled sentiment value of this document.

$$W = \{w_1, w_2, w_3, \ldots, w_n\} \tag{1}$$

Then, choose a set S, such that;

$$S \subset W \quad \wedge \quad \sum_{i=1}^{n} S_i = \tau \quad (or \quad closest) \tag{2}$$

Where S_i is the sentiment score of i-th token in subset S. Then, we introduce a vector $\vec{x}$ as a feasible solution.

$$Feasible \quad Solution : \vec{x} = (x_1, x_2, x_3, \ldots, x_n) \tag{3}$$

$$where \quad x_i \in \{0, 1\}$$

$$\sum_{i=1}^{n} w_i x_i = \tau \quad (or \quad closest) \tag{4}$$

$$for \quad i = 1, 2, \ldots, n.$$

$$Objective \quad Function : P(\vec{x}) = \sum_{i=1}^{n} w_i.x_i = \tau \tag{5}$$

$$where \quad \vec{x} = (x_1, x_2, x_3, \ldots, x_n)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

3) Fitness calculation

In order to apply GA to this problem, the binary string $\vec{x}$ is to be chosen as genotype. The fitness function $f(\vec{x})$ which is a simple form of our objective function $P(\vec{x})$, is a vital part of our GA based feature selection. It determines the criteria for the best candidates which will be allowed to produce offsprings and survive in the next generation. We designed our fitness function in a way that it should make the solution converge right from the first generation. The fitness function to evaluate the fitness of a selected set of features depends on the relative distance from the labeled sentiment value where distance is in terms of polarity score determined by SentiWordNet lexical database. The lesser the polarity distance between the class label and calculated score, the most feasible is the current solution, hence, more probable to survive in the next generation. The fitness function $f(\vec{x})$ to evaluate the fitness of each individual genotype is described in equation (6). The fitness calculation is also described in Algorithm 3.

$$FitnessFunction: f(\vec{x}) = s.\big(\tau - P(\vec{x})\big) + \big(1 - s\big).P(\vec{x}) \tag{6}$$

where,

$$s = \begin{cases} 1 & \text{if } \big(\tau - P(\vec{x})\big) = 0 \quad (or \quad min.), \quad \vec{x} \text{ is feasible} \\ 0 & \text{otherwise} \end{cases}$$

Input: `T`: list of tokens
`G`: current genotype
`S`: labelled sentiment
Output: `score`: polarity score
$sum \leftarrow 0$
foreach $g_i \in G$ **do**
  `/* 1 means include, 0 means exclude. Calculate polarity score of only subset of T determined by G */`
  if $g_i = 1$ **then**
    $TT_i \leftarrow getPosTag(T_i)$
    $score \leftarrow getSentiWordnetScore(T_i, TT_i)$
    $sum+ = score$
end
$score \leftarrow (S - sum)$
return *score*

Algorithm 3: FITNESS_CALCULATION

4) Algorithm and Analysis

The algorithm for GA based feature selection is shown in Algorithm 4. [29] [30] [31] We run the simulation until `N` number of generations so that entire population should converge to a single optimal solution. In each generation, different steps that constitute the working of the GA are performed. This includes crossover, mutation, offspring generation, and fitness evaluation. These processes of a single generation are described in Algorithm 5.

Input: A finite list $A = \{a_1, a_2, \ldots, a_n\}$ of tokens and a labelled sentiment value T.
Output: a list of optimal features
Let P be the initial randomly seeded population and k be the number of generations
$numGenerations \leftarrow k$
$count \leftarrow 0$
while *count* <*numGenerations* **do**
  $ProduceNextGeneration(P, A, T)$
end
return P_0

Algorithm 4: FEATURE_SELECTION_GA

Input: Initial population P, A and target T
$P_n \leftarrow \phi$
Let P_n be the new population.
while $P_n.size$ <*P.size* **do**
  Let i, j, k and l be 4 distinct random integers.
  Choose 4 chromosomes $ch1$, $ch2$, $ch3$, $ch4$ at these random indices from P.
  Check the fitness between $ch1$ and $ch2$, and between $ch3$ and $ch4$ and let the winners be two parents.
  $w1 \leftarrow winner_{12}$
  $w2 \leftarrow winner_{34}$
  Perform *uniform crossover* on $w1$ and $w2$ with *probability* 0.5 and generate 2 new children *child1* and *child2*.
  $Prob_{mutate} \leftarrow 0.01$
  $r \leftarrow random()$
  if $r < prob_{mutate}$ **then**
    $k \leftarrow random(child1.size)$
    if $child1(k) = 1$ **then**
      $child1(k) \leftarrow 0$
    else
      $child1(k) \leftarrow 1$
    end
    $k \leftarrow random(child2.size)$
    if $child2(k) = 1$ **then**
      $child2(k) \leftarrow 0$
    else
      $child2(k) \leftarrow 1$
    end
  end
  $isChild1Good \leftarrow child1.CalculateFitness()$ *is better than* $w1.CalculateFitness()$
  $isChild2Good \leftarrow child2.CalculateFitness()$ *is better than* $w2.CalculateFitness()$
  if $isChild1Good$ **then**
    $P_n.add(child1)$
  else
    $P_n.add(w1)$
  end
  if $isChild2Good$ **then**
    $P_n.add(child2)$
  else
    $P_n.add(w2)$
  end
end
$P \leftarrow P_n$
return

Algorithm 5: GENERATE_NEXT_GEN_GA

The time complexity of GA depends upon the fitness function. There were two ways to implement this problem, either keep generating new population until the solution is achieved or fix the number of generations to a big enough number k such that the solution can converge before reaching that limit. For the sake of simplicity, the latter one is used and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

the limit is set to k generations. Let N_p and N_a be the size of the population and the size of the keyword list respectively. In this case, the value of k is 5000 and the value of N_p is 40. This is just the initial capacity and the list will be recreated to accommodate new elements. In Algorithm 4, the outer while loop runs until the k number of generations and for each iteration, it calls $GenerateNextGenGA$. In Algorithm 5, the while loop at $line-2$ runs until the N_p. In this while loop, there are other loops that iterate over each gene g_j of each chromosome P_i in operations like $crossover$ and $fitness$ (they are not included in the above algorithms for the sake of clarity). Thus these for loops iterate until the size of each chromosome N_c which is as equal to N_a. So the complexity of each evolution of GA is;

$$T(n) = k * N_p * N_c \tag{7}$$

$$T(n) = O(N_p * N_c) \tag{8}$$

by ignoring the constant value and all the lower order terms. Please note that this time complexity is subject to fixing the number of generations to some constant k.

IV. RESULTS AND DISCUSSION

This section presents our results and the discussion. We first describe our software and hardware setup for evaluations. Later, we explain different evaluation parameters and discuss the performance of our system on these parameters. We use several performance metrics i.e., precision, recall, F-measure, and execution time. We further discuss the comparison of different ML classifiers e.g., NB, J48, PART, SMO, IB-k, and JRip. Finally, we also discuss the relative performance of our framework using three different approaches to sentiment analysis; SentiWordnet alone, ML, hybrid method with GA optimized feature-set.

A. EXPERIMENTAL SETUP

In order to evaluate different approaches to sentiment analysis used in the proposed framework, we use three datasets: UCI ML dataset for sentiment scoring [15] which consist of user's reviews and their relevant sentiment scores from three different websites, Twitter labeled sentiment analysis dataset [16], and geopolitical dataset related to 2016 United States Presidential Election [17].

The experiments are performed on a desktop computer with Core i7 processor having 2.6 GHz frequency, 8GB of RAM, and 1TB of hard disk space. The development of the framework is carried out in Java language with Eclipse IDE as the workbench.

B. REVIEWS DATASET

In this section, we discuss each experiment, relevant graphs, and performance metrics on reviews dataset from UCI ML repository. This dataset contains 3000 instances, labeled with 1 for positive or 0 for negative sentiment. These reviews were collected randomly from three different larger review sources: movie reviews from IMDB, restaurant reviews from Yelp, and product reviews from Amazon [32].

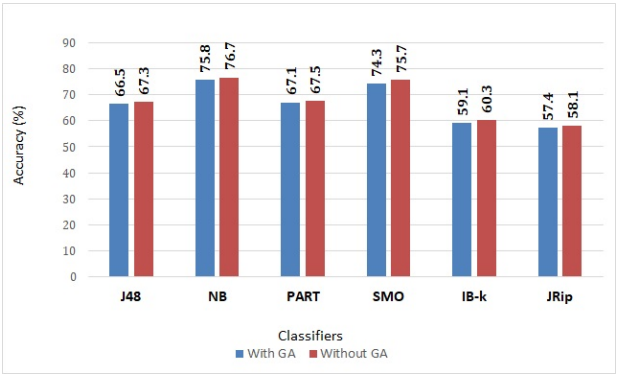

FIGURE 3. Accuracy comparison of two feature selection techniques on IMDB movies reviews

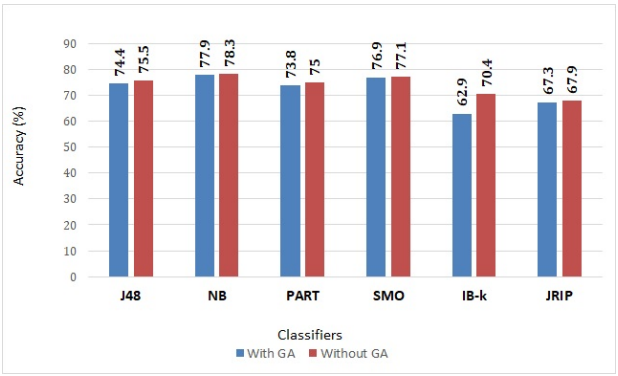

FIGURE 4. Accuracy comparison of two feature selection techniques on Amazon product reviews

1) Features size and Accuracy comparison of ML approaches

First of all, we perform an accuracy comparison of GA based and non-GA based ML approaches for reviews from three different resources. Figure 3, Figure 4, and Figure 5 show the comparison of both ML techniques on IMDB, Amazon, and Yelp reviews respectively using six different classifiers. As we observe that the accuracy of GA optimized reduced feature-set results in almost equal to non-GA based technique (which contains 40% more features). We also observe that out of these six classifiers, NB and SMO show close to 80% accuracy in both approaches.

In order to substantiate our claim that the GA based approach for optimal feature selection results in significant feature size reduction while maintaining the similar accuracy, we perform a feature size comparison experiment. Figure 6 shows the size of the feature-set before and after we performed GA optimization on feature selection. As we can see that GA optimization has reduced the feature size by almost 40% which is significant. An important point to note is that we have already seen in Figures 3, 4, and 5 that accuracy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

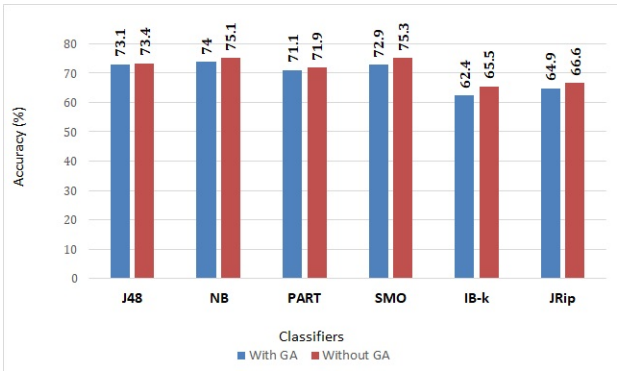

FIGURE 5. Accuracy comparison of two feature selection techniques on Yelp restaurants reviews

FIGURE 6. Features size comparison of feature vector before and after using GA optimization

of both approaches is same but GA optimization gives us a reduced feature size. This has a significant impact on the scalability of the system. Using bag-of-words as feature-set can result in a huge bottleneck when using larger dataset.

Figure 7 shows the scalability of our system in terms of execution time on GA optimized ML approach for sentiment analysis. It also shows the parts of execution and how much time is spent during each step. As we observe that GA take almost 60%-70% of the total execution time. However, our basic assumption is to optimize space to achieve better scalability at the cost of execution time. The execution time spent in GA operations can be reduced by employing different parallelization techniques, however, these approaches are beyond the scope of the current state of this paper.

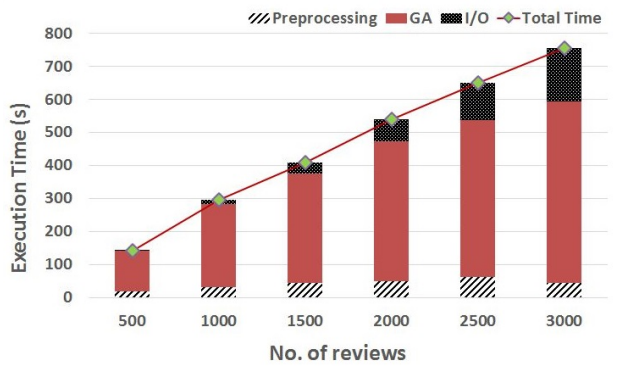

FIGURE 7. Scalability and Time consumption of different steps

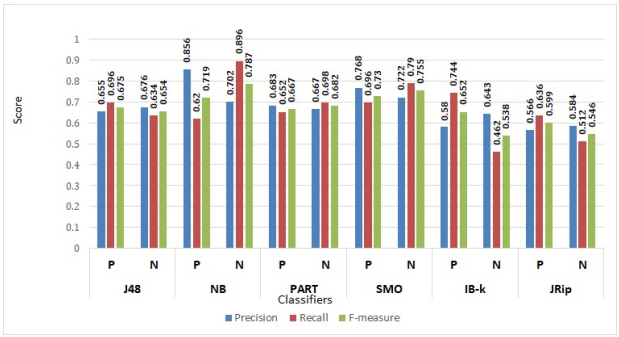

FIGURE 8. Precision, Recall, and F-measure comparison of six different classifiers using GA optimized features on IMDB dataset

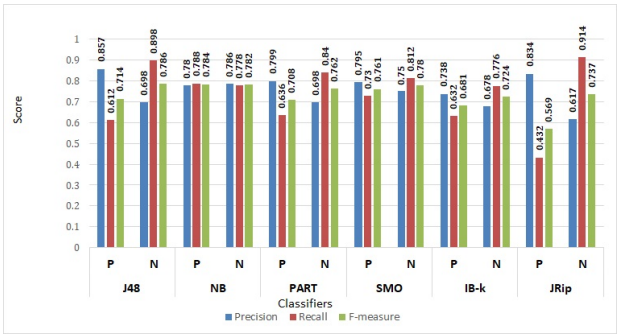

FIGURE 9. Precision, Recall, and F-measure comparison of six different classifiers using GA optimized features on Amazon dataset

2) Comparing ML classifiers under GA optimization

We perform a relative comparison of six different ML classifiers by using GA enhanced feature-set. We use precision, recall, and F-measure as the performance metric for classifiers. The metrics are calculated for both positive "P" and negative "N" classified documents on each classifier. Furthermore, we perform the same tests for reviews from three different resources which we have discussed earlier. The results on IMDB dataset is shown in Figure 8. We observe that the NB classifier has the highest recall with 0.89 for negative class, followed by SMO with 0.8 for negative class. This is in alignment with our previous results of accuracy comparison of ML classifiers which shows that NB has the highest accuracy under GA. For precision, we observe a similar trend as NB for the positive class has the highest precision of 0.85, followed by SMO with 0.77. For F-measure, NB for negative has the highest F-measure with 0.787 while SMO for negative with 0.755 closely followed. We observe that for negative class, NB has the highest values while SMO came to be the second best while for the positive class, SMO has the highest F-Measure of 0.73 with closely followed by NB with F-measure of 0.719.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

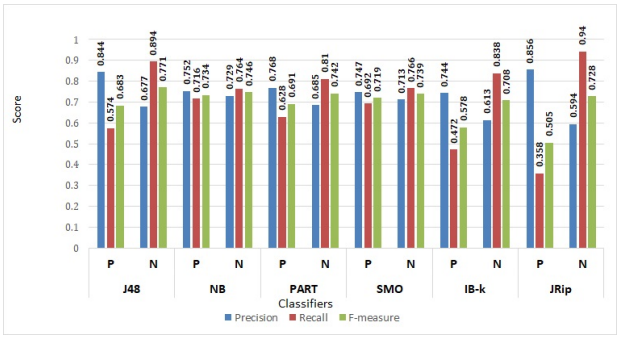

FIGURE 10. Precision, Recall, and F-measure comparison of six different classifiers using GA optimized features on Yelp datasets

IB-k results in 0.652 F-measure for positive whereas 0.538 for the negative class. JRip shows the least score with 0.599 F-measure for positive class while 0.546 for the negative class.

The results of other two dataset are shown in Figure 9 and Figure 10. For Amazon dataset, we observe that JRip has the highest recall of 0.914 closely followed by J48 with 0.89 (both on negatively classified documents). Similarly, J48 has the highest precision for positively classified documents closely followed by JRip for positive, but their F-measure is affected by low recall. For F-measure, NB for positive has the highest F-measure with 0.784 closely followed by NB for negative with 0.782.

Lastly, we evaluate the Yelp dataset. For recall, JRip for negative has the highest recall with 0.94 while IB-k with negative class closely followed. For precision, JRip for the positive class has the highest value as 0.856 followed by J48 on positive class with 0.844. For F-measure, J48 for negatives class has the highest F-measure with 0.771 followed by NB for negative class with 0.74.

We found that in all these results, overall performance is better in NB classifier while SMO and J48 closely followed. Since we will be evaluating three different approaches to sentiment analysis, we will be using NB because it previously showed the best accuracy as compared to other classifiers.

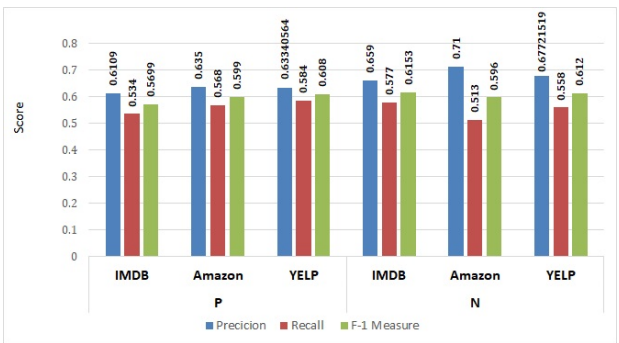

FIGURE 11. Precision, Recall, and F-measure of SentiWordNet based sentiment analysis on IMDB, Amazon, and Yelp reviews

3) Comparing three different approaches to sentiment analysis

As discussed before, we include three different approaches of sentiment analysis in our framework. First, we use only SentiWordNet (SWN) ontology to find polarity score of each keyword (after pos-tagging) and then aggregate the overall score of a document to find whether the overall notion is positive or negative. The second approach is based on ML in which we use a feature-set and different classifiers (as discussed in previous sections) to classify positive and negative documents. However, this approach is further divided into two approaches based on how the feature selection is performed. The first approach in ML technique uses bag-of-words as feature vector which means all the keywords having any polarity score attached are included in the feature vector. The second approach in ML technique uses a GA based optimized feature selection. In this approach, each document is modeled onto the GA model and GA simulation is run for several hundred generations to find the optimal set of features which results in best sentiment classification.

We evaluate all three approaches (SWN, ML with all features, and ML with GA optimized features) using precision, recall, F-measure as performance metrics. In order to provide a more detailed analysis, we perform these evaluations on three different reviews datasets e.g., IMDB, Amazon, Yelp which we have discussed before. Finally, we also evaluate the accuracy of these three approaches on all three datasets. An important point to note is that the results for ML approach are taken only using NB classifier. We have already seen in the previous discussion that NB gave best results for ML on both GA and non-GA approach.

Figure 11 shows the results of sentiment analysis only using SWN polarity scoring and aggregation. We observe that Amazon dataset for the negative class shows the highest precision score of 0.71 which is closely followed by Yelp dataset for the negative class with a score of 0.67. Similarly, Yelp dataset for the positive class has the highest recall of 0.58 while IMDB dataset for the negative class came afterward with 0.577. For F-measure, we found that IMDB for the negative class has the highest score of 0.615 followed by Yelp for positive class with a score of 0.608.

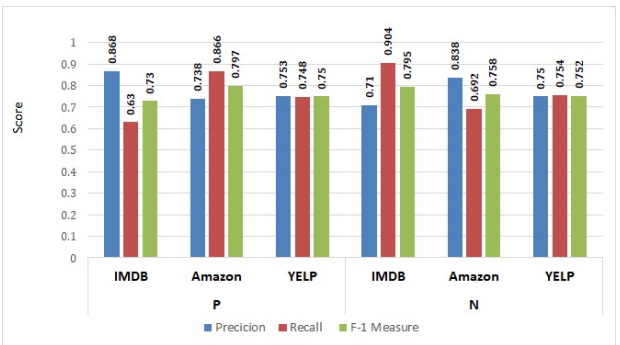

FIGURE 12. Precision, Recall, and F-measure of simple feature selection for sentiment analysis on NB classifier

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

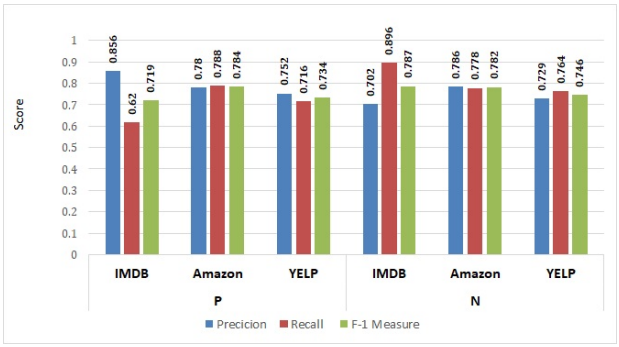

FIGURE 13. Precision, Recall, and F-measure of GA optimized feature selection for sentiment analysis on NB classifier

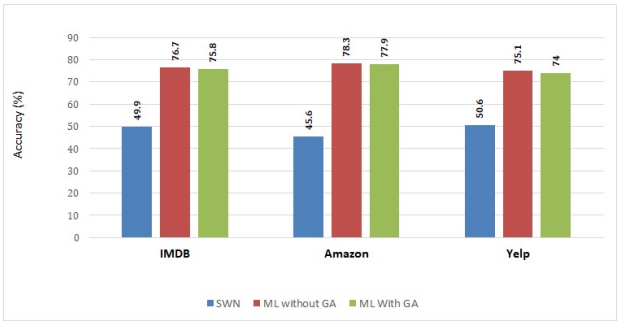

FIGURE 14. Accuracy comparison of three main approaches to sentiment analysis on IMDB, Amazon, and Yelp reviews

The results for the second approach which was ML using bag-of-words as features are shown in Figure 12. We observe that IMDB for the positive class has the highest precision of 0.868 followed by Amazon for negative class having 0.838. For recall, IMDB for negative shows the highest recall with 0.904 score which is followed by Amazon for positive class with a score of 0.866. Similarly, Amazon for positive class shows the highest F-measure of 0.80 and IMDB for negative comes afterward with a slightly lower score of 0.79.

The results of the third approach with optimized feature selection using GA are shown in Figure 13. We observe that IMDB with positive class has the highest precision of 0.86 followed by Amazon for negative class with a score of 0.786. Similarly, IMDB for the negative class has the highest recall of 0.896 while Amazon for the positive class came afterward with a score of 0.788. For F-measure, we found IMDB for the negative class has the highest score of 0.787 followed by Amazon for negative class with a slightly lower score of 0.782.

Finally, we compare the accuracy of these three approaches used for sentiment analysis in our framework. The accuracy comparison is shown in Figure 14. As we can see that SWN approach has almost 50% accuracy at best which is not feasible for real-time analysis. The ML approaches (with and without GA optimization) has an accuracy ranging from 74% to 78%. However, GA optimized ML technique has 40% reduced feature-set. Overall, we see that GA optimized ML

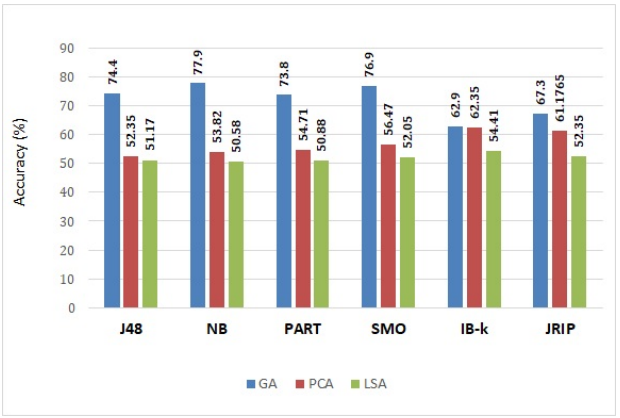

FIGURE 15. Accuracy comparison of GA based reduction with PCA and LSA based reduction techniques on Amazon dataset

in case of Amazon dataset has the highest accuracy of 77.9% while non-GA based ML has around 78.3%.

We conclude that, in sentiment analysis, GA based optimal feature selection does not much affect the accuracy as compared to non-GA based feature selection. At the same time, GA based approach has also reduced the feature-set size by a marginal 40% as compared to other approaches.

4) Comparison of GA with PCA and LSA

In the final part of our evaluation, we demonstrate that our GA based feature reduction techniques perform better than PCA and LSA based feature reduction. Figure 15 shows the accuracy graph of all three feature reduction techniques on Amazon dataset. As we can see that GA based approach has, on average, 15.38% better accuracy than PCA and 20.29% better accuracy than LSA. Similarly, we observe that the NB classifier shows the highest accuracy difference with 24.08% increase than PCA, and 27.32% increased performance than LSA based feature reduction. Based on these observations, we conclude that our GA based technique is more effective than two of the existing well-known approaches for feature reduction.

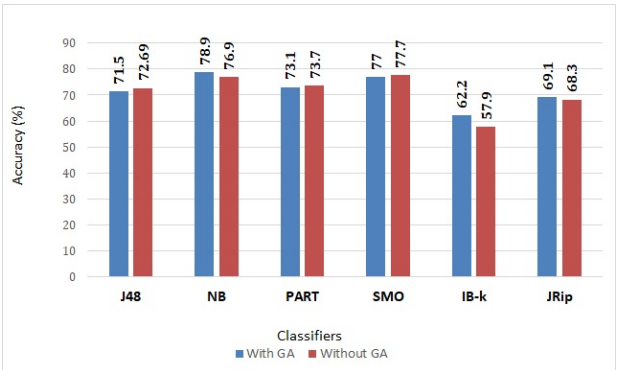

FIGURE 16. Accuracy comparison of two feature selection techniques on tweets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

C. TWITTER DATASET

In this section, we discuss the experiment, relevant graphs, and performance metrics on twitter dataset.

1) Features size and Accuracy comparison of ML approaches

First of all, we perform an accuracy comparison of GA based and non-GA based ML approaches. Figure 16 shows the comparison of both ML techniques on twitter dataset using six different classifiers. As we observe that the accuracy of GA optimized reduced feature-set results in almost equal to the non-GA based technique (which contains 42% more features). We also observe that out of these six classifiers, NB and SMO show close to 80% accuracy in both approaches. On our Twitter dataset, in case of IB-k, NB, and JRip, the GA based feature reduction technique results in almost 4.3%, 2%, and 0.8% respectively better accuracy than the non-GA based feature selection.

FIGURE 17. Scalability and Time consumption of different steps on Twitter dataset

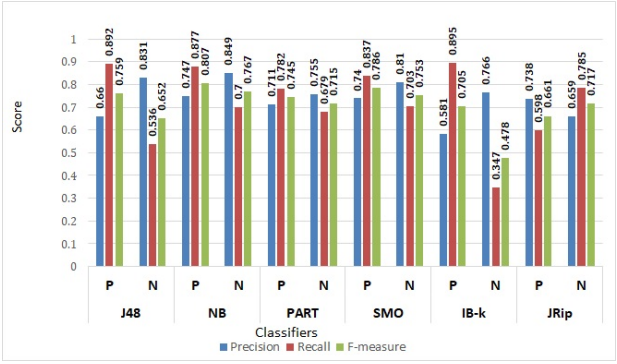

FIGURE 18. Precision, Recall, and F-measure comparison of six different classifiers using GA optimized features on tweets

For Twitter dataset, the number of features before performing GA are 2722 and after performing GA optimization on feature-set, feature size decreases to 1562 which is almost 42% reduction in the size. We have already seen in Figure 16 that accuracy of both approaches is same or even better in case of IB-k, NB, and JRip, and GA optimization gives us a reduced feature-set size. This has a significant impact on the scalability of the system.

Figure 17 shows the scalability of our system in terms of execution time on GA optimized ML approach for sentiment analysis. It also shows the parts of execution and how much time is spent during each step. As we observe that preprocessing takes almost 55% - 60% of the total execution time. This is because twitter data is noisy and requires more cleaning before being able to process for ML. After preprocessing, GA also consumes a lot of time, however, our basic assumption is to optimize space to achieve better scalability at the cost of the execution time. As we have discussed earlier that the execution time spent in GA operations can be reduced by employing different parallelization techniques.

2) Comparing ML classifiers under GA optimization

Figure 18 shows a relative comparison of six different ML classifiers by using GA enhanced feature-set. We observe that IB-k classifier has the highest recall with 0.895 for positive classified tweets, followed by J48 with 0.892 but both have a very low precision for positive class. For positive class, the precision of NB i.e.0.747 is the highest and same for the negative class with the precision of 0.849. NB also have the highest F-measure for both the positive class with 0.807 closely followed by SMO with 0.786 and for the negative class with 0.767 followed by SMO with 0.753.

We found that overall performance is better on NB classifier while SMO closely followed which is in accordance with our previous results on reviews dataset.

3) Comparing three different approaches to sentiment analysis

Our framework contains three different approaches to sentiment analysis. We have already seen in the previous discussion that NB gave best results for ML on both GA and non-GA approach, so, Figure 19 shows the accuracy comparison of these three approaches of sentiment analysis

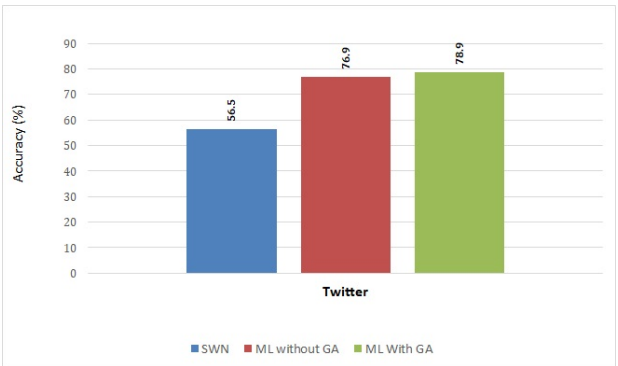

FIGURE 19. Accuracy comparison of three main approaches to sentiment analysis on Twitter dataset

As we can see that SWN approach has almost 56% accuracy at best which is not feasible for real-time analysis. Overall, we see that GA optimized ML has the highest accuracy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

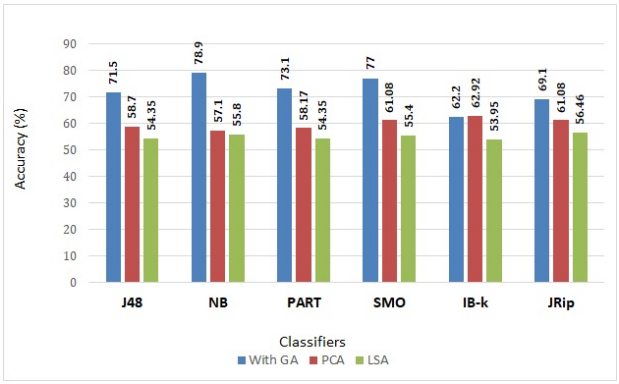

FIGURE 20. Accuracy comparison of GA based reduction with PCA and LSA based reduction techniques

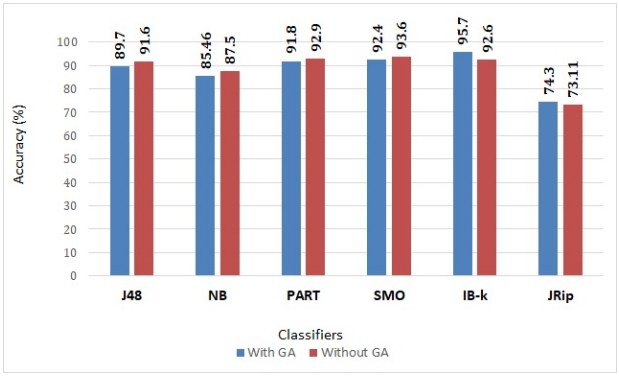

FIGURE 21. Accuracy comparison of two feature selection techniques using geopolitical data

of almost 79% while non-GA based ML has around almost 77%. GA based optimal feature selection has improved the accuracy and at the same time, GA based approach has also reduced the feature-set size by a marginal 43%.

4) Comparison of GA with PCA and LSA

Figure 20 shows the accuracy graph of all three feature reduction techniques on Twitter Sentiment dataset. As we can see that GA based approach has, on average, 10.4% better accuracy than PCA and 14.5% better accuracy than LSA. All the classifiers show better accuracy on GA based feature-set except IB-K which shows better accuracy with PCA based feature-set. We observe that the NB classifier shows the highest accuracy difference with 21.8% increase than PCA, and 23.1% increased performance than LSA based feature reduction. Hence, it proves that our GA based technique is more effective than two of the existing well-known approaches for feature reduction.

D. GEOPOLITICAL DATASET

In this section, we discuss the experiment, relevant graphs, and performance metrics on a geopolitical dataset which is related to the 2016 United States Presidential Election. This dataset contains tweet IDs collected using candidates and key

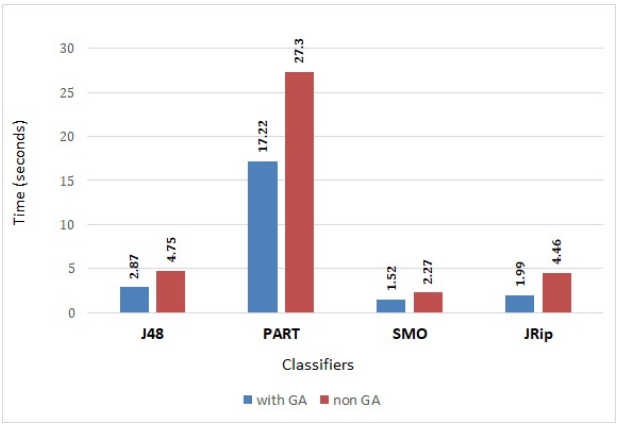

FIGURE 22. Time comparison for models training of two feature selection techniques

election hashtags. We use Hydrator [33], a desktop application that takes in tweet IDs and returns the corresponding data from Twitter as JSON. We selected tweets related to first debate for our case study. To label this dataset, we use emoticons methods as used by several researchers [34] [35] using emoticons selected by Hu et al. [36]. To test on the proposed framework, we randomly selected almost 1000 tweets which discuss multiple topics related to US government first debate. This dataset can be used to measure public sentiments and views regarding various topics which have applications in various areas including security and surveillance, law-and-order, and public administration.

1) Features size and Accuracy comparison of ML approaches

First of all, we perform an accuracy comparison of GA based and non-GA based ML approaches. Figure 21 shows the comparison of both ML techniques on the geopolitical dataset using six different classifiers. As we observe that the accuracy of GA optimized reduced feature-set results in almost equal to non-GA based technique (which contains 34% more features). On geopolitical dataset, we observe that out of these six classifiers, IB-k and SMO show more than 90% accuracy in both approaches and in case of IB-k and JRip, GA based technique show 3.1% and 1.19% respectively better accuracy than non-GA based method even with decreased feature-set. We observe that IB-k classifier with GA based approach outperforms all the classifiers with the accuracy of 95.7%.

Number of features before performing GA was 1899 and after performing GA optimization on feature-set, feature size decreased to 1246 which is almost 34% reduction in size. As we have stated earlier that this reduced features set have a significant impact on the scalability of the system which is shown in Figure 22. This figure shows a comparison of time required to build the ML models. We can see that applying GA based feature selection significantly decreases the time required to build the model. In the case of PART which is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2892852, IEEE Access

slowest among all the classifiers, using GA based technique, we are able to reduce time by 37%. Largest speedup is achieved with JRIP which is 55%. Values of time for NB and IB-k are so small to be shown on the graph so we exclude them. For time required to preprocess and apply GA on data, we found the same patterns as shown in Figure 17.

2) Comparing ML classifiers under GA optimization

Figure 23 shows a relative comparison of six different ML classifiers by using GA enhanced feature-set. We observe that IB-k classifier has the highest recall with 0.978 for positive class and highest precision of 0.975 for negative class. This results in overall highest F-measure of IB-k classifier for both positive and negative class and hence highest accuracy among all the six classifiers. JRIP also have the good recall measures i.e. 0.944 for positive class but low precision affects its F-measure. SMO shows the second highest F-measure for both positive and negative class. We found that overall performance is better in IB-k classifier while SMO closely followed.

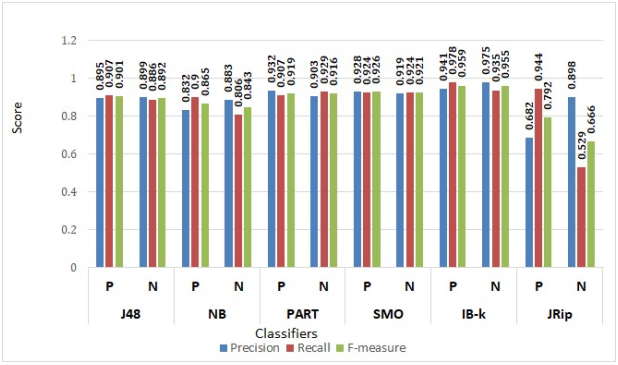

FIGURE 23. Precision, Recall, and F-measure comparison of six different classifiers using GA optimized features on tweets

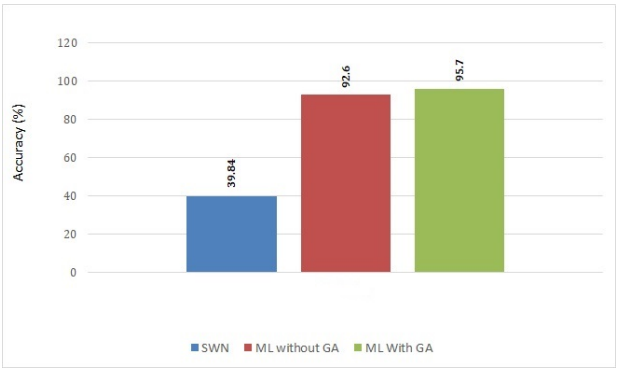

FIGURE 24. Accuracy comparison of three main approaches to sentiment analysis on geopolitical dataset

3) Comparing three different approaches to sentiment analysis

Our framework contains three different approaches to sentiment analysis. On geopolitical dataset, we found that IB-k classifier gives best results on both GA and non-GA based approach. It is also shown in Figure 21 that GA based IB-k gives higher accuracy than non-GA based IB-k. So, we choose IB-K classifier to compare our three approaches. Figure 24 shows the accuracy comparison of three approaches of sentiment analysis for IB-k classifier.

As we can see that SWN approach has almost 39.84% accuracy at best which is not feasible for practical application. Overall, we see that GA optimized ML has the highest accuracy of almost 95.7% while non-GA based ML has around almost 92.6%. GA based optimal feature selection has improved the accuracy and at the same time, GA based approach has also reduced the feature-set size by a marginal 34%.

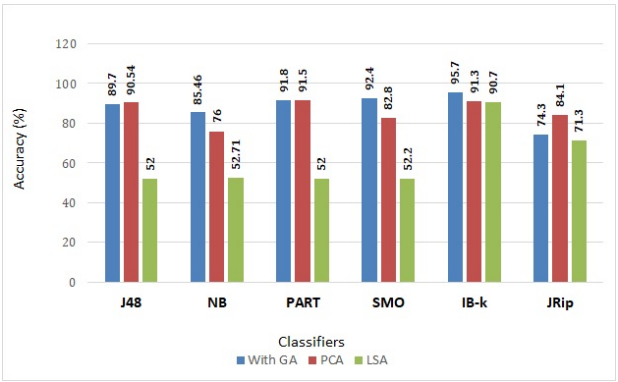

FIGURE 25. Accuracy comparison of GA based reduction with PCA and LSA based reduction techniques on geopolitical dataset

4) Comparison of GA with PCA and LSA

Figure 25 shows the accuracy graph of all three feature reduction techniques on the geopolitical dataset. All the classifiers show better accuracy on GA based feature-set except J48 and JRip which show better accuracy with PCA based feature reduction dataset. As on geopolitical dataset, IB-k has the highest accuracy which also shows 4.4% better accuracy as compared to PCA and 5% better accuracy as compared to LSA when GA based feature-set is used. We observe that the SMO classifier shows the highest accuracy difference with 9.6% increased performance than PCA and 40.2% increased performance than LSA based feature reduction. This also proves that our GA based technique is more effective than two of the existing well-known approaches for feature reduction.

V. CONCLUSION

In this paper, we have presented the design, development, and evaluation of our integrated sentiment analysis framework in detail. We employed three different approaches to sentiment analysis which includes SWN, ML, and ML with GA optimized feature selection. We proposed and developed an evolutionary model for feature selection using GA's evolutionary model. This novel approach resulted in 36% - 43% reduced feature size and about 5% increased efficiency as

compared to a normal ML approach. We also presented a detailed evaluation of these approaches with respect to different datasets. Furthermore, our detailed analysis of different ML classifiers revealed that the NB classifier has the highest accuracy (about 80%) while using our GA based optimal feature selection on Twitter and reviews dataset while in case of the geopolitical dataset, IB-k outperformed all the classifiers with the accuracy of 95%.

Furthermore, we evaluated our proposed technique for scalability by using execution time comparison. We found that our system showed a linear speedup with the increased dataset size. Although, the time spent in the selection of optimal feature-set using GA took about 60% to 70% of the total execution time on reviews dataset, however, it still remained linear and produced a feature-set with 40% reduced size than the original feature-set. GA based feature set results in a speedup of modeling the classifiers up to 55%

In order to demonstrate the benefit of using our feature reduction algorithm over other feature reduction techniques, we have provided an accuracy comparison of GA based hybrid approach with PCA and LSA. The results showed that our GA based feature reduction showed up to 15.4% increased accuracy over PCA and up to 40.2% increased accuracy over LSA. This strengthens our claim that our proposed algorithm is fast, accurate, and scales well as the dataset grows bigger.

We conclude that our sentiment analysis framework has proved to be a great addition in the discipline of opinion mining. It provided the flexibility of choosing among three widely used sentiment analysis techniques according to custom needs. With additional benefits of GA based optimization, it reduces feature size and improves efficiency while maintaining the scalability. In the future, we aim to extend this framework for cyber-intelligence so that it would help generate recommendations for law-enforcement agencies based on user opinions.

ACKNOWLEDGMENT

This study is partially supported by Research Incentive Fund (R15048) and Research Clusters (R17082 & R16083), Zayed University, United Arab Emirates.

REFERENCES

[1] P. DiMaggio, E. Hargittai, W. R. Neuman, and J. P. Robinson, "Social implications of the internet," Annual review of sociology, pp. 307–336, 2001.

[2] C. Wang and P. Zhang, "The evolution of social commerce: The people, management, technology, and information dimensions," Communications of the Association for Information Systems, vol. 31, no. 5, pp. 1–23, 2012.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and trends in information retrieval, vol. 2, no. 1-2, pp. 1–135, 2008.

[4] A. Davies and Z. Ghahramani, "Language-independent bayesian sentiment mining of twitter," in Workshop on Social Network Mining and Analysis, 2011.

[5] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach," Journal of Informetrics, vol. 3, no. 2, pp. 143–157, 2009.

[6] A. Collomb, C. Costea, D. Joyeux, O. Hasan, and L. Brunie, "A study and comparison of sentiment analysis methods for reputation evaluation," Rapport de recherche RR-LIRIS-2014-002, 2014.

[7] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual web texts," Information retrieval, vol. 12, no. 5, pp. 526–558, 2009.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," Computational linguistics, vol. 37, no. 2, pp. 267–307, 2011.

[9] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in Proceedings of the 2008 International Conference on Web Search and Data Mining. ACM, 2008, pp. 231–240.

[10] R. Feldman, "Techniques and applications for sentiment analysis," Communications of the ACM, vol. 56, no. 4, pp. 82–89, 2013.

[11] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in Proceedings of LREC, vol. 6. Citeseer, 2006, pp. 417–422.

[12] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," Chemometrics and intelligent laboratory systems, vol. 2, no. 1, pp. 37–52, 1987.

[13] S. T. Dumais, "Latent semantic analysis," Annual review of information science and technology, vol. 38, no. 1, pp. 188–230, 2004.

[14] S. Goel, "Cyberwarfare: connecting the dots in cyber intelligence," Communications of the ACM, vol. 54, no. 8, pp. 132–140, 2011.

[15] UCI, "Uci ml repository - sentiment analysis dataset," 2015, accessed: 2018-06-08. [Online]. Available: [dataset] http://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

[16] J. A. Bowden, "Twitter sentiment analysis," 2016, accessed: 2018-06-08. [Online]. Available: [dataset] https://old.datahub.io/dataset/twitter-sentiment-analysis

[17] J. Littman, L. Wrubel, and D. Kerchner, "2016 united states presidential election tweet ids," 2016, accessed: 2018-12-21. [Online]. Available: [dataset] https://doi.org/10.7910/DVN/PDI7IN

[18] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093–1113, 2014.

[19] A. Khan, B. Baharudin, and K. Khairullah, "Sentiment classification using sentence-level lexical based semantic orientation of online reviews," Trends in Applied Sciences Research, vol. 6, no. 10, pp. 1141–1157, 2011.

[20] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in Proceedings of the Workshop on Languages in Social Media. Association for Computational Linguistics, 2011, pp. 30–38.

[21] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" ICWSM, vol. 11, pp. 538–541, 2011.

[22] M. Pontiki, H. Papageorgiou, D. Galanis, I. Androutsopoulos, J. Pavlopoulos, and S. Manandhar, "Semeval-2014 task 4: Aspect based sentiment analysis," in Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 27–35.

[23] P. C. S. Njolstad, L. S. Hoysaeter, W. Wei, and J. A. Gulla, "Evaluating feature sets and classifiers for sentiment analysis of financial news," in Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on, vol. 2. IEEE, 2014, pp. 71–78.

[24] M. Govindarajan, "Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm," IJACR, vol. 3, no. 4, pp. 139–145, 2013.

[25] A. McCallum, "Rainbow stopwords," 1998, accessed: 2018-06-08. [Online]. Available: http://www.cs.cmu.edu/ mccallum/bow/rainbow/

[26] M. McCandless, E. Hatcher, and O. Gospodnetic, Lucene in Action: Covers Apache Lucene 3.0. Manning Publications Co., 2010.

[27] M. F. Porter, "An algorithm for suffix stripping," Program, vol. 14, no. 3, pp. 130–137, 1980.

[28] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[29] L. M. Schmitt, "Theory of genetic algorithms," Theoretical Computer Science, vol. 259, no. 1-2, pp. 1–61, 2001.

[30] M. Mitchell, An introduction to genetic algorithms. MIT press, 1998.

[31] D. Beasley, D. R. Bull, and R. R. Martin, "An overview of genetic algorithms: Part 1, fundamentals," University computing, vol. 15, no. 2, pp. 56–69, 1993.

[32] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, "From group to individual labels using deep features," in Proceedings of the 21th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, pp. 597–606.

[33] "Hydrator," 2016, accessed: 2018-12-21. [Online]. Available: [dataset] https://github.com/DocNow/hydrator

[34] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in LREc, vol. 10, no. 2010, 2010, pp. 1320–1326.

[35] A. Bifet and E. Frank, "Sentiment knowledge discovery in twitter streaming data," in International conference on discovery science. Springer, 2010, pp. 1–15.

[36] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," in Proceedings of the 22nd international conference on World Wide Web. ACM, 2013, pp. 607–618.